Network Working Group
Request for Comments: 995

ANSI X3S3.3 86-118
ISO TC97/SC6/N 4053
April 1986

I S O
INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION

ISO/TC 97/SC 6
TELECOMMUNICATIONS AND INFORMATION
EXCHANGE BETWEEN SYSTEMS
Secretariat: USA (ANSI)

Title: End System to Intermediate System Routing Exchange Protocol for use in conjunction with ISO 8473

Source: SC6/WG2
Project 97.6.41

This document is a progression of SC6/N3862, edited to incorporate member body comments and discussion at the Florence meeting of SC6/WG2. Pursuant to Recommendation 5 of that meeting, comments from member bodies on this revision text are requested for discussion at the Tokyo meeting of SC6 and WGs.

Contents

1 Introduction 5

2 Scope and Field of Application 6

3 References 7

SECTION ONE. GENERAL 9

4 Definitions 9
4.1 Reference Model Definitions . . . . . 9
4.2 Network Layer Architecture Definitions . . . . . 9
4.3 Network Layer Addressing Definitions . . . . . 9
4.4 Local Area Network Definitions . . . . . 10
4.5 Additional Definitions . . . . . 10

5 Symbols and Abbreviations 10
5.1 Data Units . . . . . 10
5.2 Protocol Data Units . . . . . 10
5.3 Protocol Data Unit Fields . . . . . 10
5.4 Parameters . . . . . 11
5.5 Miscellaneous . . . . . 11

6 Overview of the Protocol 11
6.1 Information Provided by the Protocol . . . . . 11
6.2 Subsets of the Protocol. . . . . 12
6.3 Addressing . . . . . 12
6.4 Underlying Service Assumed by the Protocol . . . . . 12
6.4.1 Subnetwork Addresses . . . . . 12
6.4.2 Subnetwork User Data . . . . . 13
6.5 Service Assumed from Local Environment . . . . . 13
6.6 Subnetwork Types . . . . . 14
6.6.1 Point-to-Point Subnetworks . . . . . 15
6.6.2 Broadcast Subnetworks . . . . . 15
6.6.3 General Topology Subnetworks . . . . . 16

SECTION TWO. SPECIFICATION OF THE PROTOCOL 18

7 Protocol Functions 18
7.1 Protocol Timers . . . . . 18
7.1.1 Configuration Timer . . . . . 18
7.1.2 Holding Timer . . . . . 18
7.2 Report Configuration Function . . . . . 18
7.2.1 Report Configuration by End Systems . . . . . 19
7.2.2 Report Configuration by Intermediate Systems . . . . . 19
7.3 Record Configuration Function . . . . . 20
7.4 Flush Old Configuration Function . . . . . 20
7.5 Query Configuration Function . . . . . 20

7.6 Configuration Response Function . . . . . 21
7.7 Request Redirect Function. . . . . 22
7.8 Record Redirect Function . . . . . 23
7.9 Refresh Redirect Function . . . . . 23
7.10 Flush Old Redirect Function . . . . . 24
7.11 PDU Header Error Detection . . . . . 24
7.12 Classification of Functions . . . . . 25

8 Structure and Encoding of PDUs 25
8.1 Structure . . . . . 26
8.2 Fixed Part . . . . . 26
8.2.1 General . . . . . 26
8.2.2 Network Layer Protocol Identifier . . . . . 27
8.2.3 Length Indicator . . . . . 27
8.2.4 Version/Protocol Identifier Extension . . . . . 27
8.2.5 Type Code . . . . . 28
8.2.6 Holding Time . . . . . 28
8.2.7 PDU Checksum . . . . . 28
8.3 Network Address Part . . . . . 28
8.3.1 General . . . . . 28
8.3.2 NPAI (Network Protocol Address Information) Encoding . . . . . 28
8.3.3 Source Address Parameter for ESH PDU . . . . . 29
8.3.4 Network Entity Title Parameter for ISH PDU . . . . . 29
8.3.5 Destination Address Parameter for RD PDU . . . . . 30
8.4 Subnetwork Address Part . . . . . 30
8.4.1 Subnetwork Address Parameter for RD PDU . . . . . 31
8.5 Options Part . . . . . 31
8.5.1 General . . . . . 31
8.5.2 Security . . . . . 32
8.5.3 Quality of Service Maintenance . . . . . 33
8.5.4 Priority . . . . . 33
8.6 End System Hello (ESH) PDU . . . . . 34
8.6.1 Structure . . . . . 34
8.7 Intermediate System Hello (ISH) PDU . . . . . 35
8.7.1 Structure . . . . . 35
8.8 Redirect (RD) PDU. . . . . 36
8.8.1 Structure . . . . . 36

9 Formal Description 37

10 Conformance 37

ANNEX A. SUPPORTING TECHNICAL MATERIAL 38
A.1 Use of Timers . . . . . 38
A.1.1 Example of Holding Time for Route Redirection . . . . 38
A.1.2 Example of Holding Timer for Configuration Information . . . . . 39
A.2 Refresh and timeout of Redirection information . . . . . 39
A.3 System Initialization Considerations . . . . . 40
A.4 Optimizations for Flushing Redirects . . . . . 41

List of Tables

1 Service Primitives for Underlying Service . . . . . 12
2 Timer Primitives . . . . . 14
3 Categories of Protocol Functions . . . . . 25
4 Valid PDU Types . . . . . 28

List of Figures

1 PDU Header -- Fixed Part . . . . . 27
2 Address Parameters . . . . . 29
3 ESH PDU - Network Address Part . . . . . 29
4 ISH PDU - Network Address Part . . . . . 30
5 RD PDU - Network Address Part . . . . . 30
6 ESH PDU - Address Part . . . . . 31
7 All PDUs - Options Part . . . . . 31
8 Encoding of Option Parameters . . . . . 32
9 ESH PDU Format . . . . . 34
10 ISH PDU Format . . . . . 35
11 RD PDU Format when Redirect is to an IS . . . . . 36
12 RD PDU Format when Redirect is to an ES . . . . . 37

## 1 Introduction

This Protocol is one of a set of International Standards produced to facilitate the interconnection of open systems. The set of standards covers the services and protocols required to achieve such interconnection.

This Protocol is positioned with respect to other related standards by the layers defined in the Reference Model for Open System Interconnection (ISO 7498) and by the structure defined in the Internal Organization of the Network Layer (DIS 8648). In particular, it is a protocol of the Network Layer. This protocol permits End Systems and Intermediate Systems to exchange configuration and routing information to facilitate the operation of the routing and relaying functions of the Network Layer.

The aspects of Network Layer routing that are concerned with communication between end systems and intermediate systems on the same subnetwork are to a great extent separable from the aspects that are concerned with communication among the intermediate systems that connect multiple subnetworks. This protocol addresses only the former aspects. It will be significantly enhanced by the cooperative operation of an additional protocol that provides for the exchange of routing information among intermediate systems, but is useful whether or not such an additional protocol is available.

This protocol provides solutions for the following practical problems:

1. How do end systems discover the existence and reachability of intermediate systems that can route NPDUs to destinations on subnetworks other than the one(s) to which the end system is directly connected?

2. How do end systems discover the existence and reachability of other end systems on the same subnetwork (when direct examination of the destination NSAP address does not provide information about the destination subnetwork)?

3. How do intermediate systems discover the existence and reachability of end systems on each of the subnetworks to which they are directly connected?

4. How do end systems decide which intermediate system to use to forward NPDUs to a particular destination when more than one intermediate system is accessible?

The protocol assumes that:

1. Routing to a specified subnetwork point of attachment address (SNPA) on the same subnetwork is carried out satisfactorily by the subnetwork itself.

2. The subnetwork is not, however, capable of routing on a global basis using the NSAP address alone to achieve communication with a requested destination.

Note:
Consequently, it is not possible to use Application Layer communication to carry out the functions of this protocol.

The protocol is connectionless, and is designed to:

1. minimize the amount of a priori state information needed by end systems before they can begin to communicate with other end systems;

2. minimize the amount of memory needed to store routing information in end systems; and

3. minimize the computational complexity of end system routing algorithms.

The protocol is also designed to operate in close conjunction with the Protocol for the Provision of the Connectionless-mode Network Service (ISO 8473). Since routing styles are usually closely related to communication styles, the information that this protocol provides to end systems and intermediate systems may or may not be appropriate information for supporting routing functions when a Network Layer protocol other than ISO 8473 is used.

## 2 Scope and Field of Application

This International Standard specifies a protocol which is used by Network Layer entities operating ISO 8473 in End Systems and Intermediate Systems (referred to herein as ES and IS respectively) to maintain routing information. The Protocol herein described relies upon the provision of a connectionless-mode underlying service.

This Standard specifies:

a) procedures for the transmission of configuration and routing information between network entities residing in End Systems and network entities residing in Intermediate Systems;

b) the encoding of the protocol data units used for the transmission of the configuration and routing information;

c) procedures for the correct interpretation of protocol control information; and

d) the functional requirements for implementations claiming conformance to this Standard.

The procedures are defined in terms of:

a) the interactions between End System and Intermediate System network entities through the exchange of protocol data units; and

b) the interactions between a network entity and an underlying service provider through the exchange of subnetwork service primitives.

This protocol does not specify any protocol elements or algorithms for facilitating routing and relaying among Intermediate Systems. Such functions are intentionally beyond the scope of this protocol.

## 3 References


ISO 7498 — Information Processing Systems --- Open Systems Interconnection - Basic Reference Model

DIS 7498/DAD1 — Information Processing Systems --- Open Systems Interconnection - Addendum to ISO 7498 Covering Connectionless-mode Transmission

ISO 8348 — Information Processing Systems --- Telecommunications and Information Exchange between Systems - Network Service Definition

ISO 8348/AD1 — Information Processing Systems --- Telecommunications and Information Exchange between Systems - Addendum to the Network Service Definition Covering Connectionless-mode Transmission

ISO 8348/AD2 — Information Processing Systems --- Telecommunications and Information Exchange between Systems - Addendum to the Network Service Definition Covering Network Layer Addressing

ISO 8473 — Information Processing Systems --- Telecommunications and Information Exchange between Systems - Protocol for Providing the Connectionless Network Service

DIS 8648 — Information Processing Systems --- Telecommunications and Information Exchange between Systems - Internal Organization of the Network Layer

| | |
|---|---|
| SC21/N965 | OSI Management Framework --- Seventh Working Draft |
| DIS 8802 | Local Area Networks |

# SECTION ONE. GENERAL

## 4 Definitions

### 4.1 Reference Model Definitions

This document makes use of the following concepts defined in ISO 7498:

(a) Network layer

(b) Network service access point

(c) Network service access point address

(d) Network entity

(e) Routing

(f) Network protocol

(g) Network relay

(h) Network protocol data unit

### 4.2 Network Layer Architecture Definitions

This document makes use of the following concepts from DIS 8648, Internal Organization of the Network Layer:

(a) Subnetwork

(b) End System

(c) Intermediate System

(d) Subnetwork Service

(e) Subnetwork Access Protocol

(f) Subnetwork Independent Convergence Protocol

### 4.3 Network Layer Addressing Definitions

This document makes use of the following concepts from DIS 8348/DAD2, Addendum to the Network Service Definition Covering Network Layer Addressing:

(a) Subnetwork address

(b) Subnetwork point of attachment

## 4.4 Local Area Network Definitions

This document makes use of the following concepts from DIS 8802, Local Area Networks:

(a) multicast address

(b) broadcast medium

## 4.5 Additional Definitions

For the purposes of this document, the following definitions apply:

Configuration: The collection of End and Intermediate Systems attached to a single subnetwork, defined in terms of the system types, NSAP addresses present, Network Entities present, and the correspondence between systems and SNPA addresses.

Network Entity Title: An identifier for a network entity which has the same abstract syntax as an NSAP address, and which can be used to unambiguously identify a network entity in an End or Intermediate System.

# 5 Symbols and Abbreviations

## 5.1 Data Units

| | |
|---|---|
| PDU | Protocol Data Unit |
| SNSDU | Subnetwork Service Data Unit |

## 5.2 Protocol Data Units

| | |
|---|---|
| ESH PDU | End System Hello Protocol Data Unit |
| ISH PDU | Intermediate System Hello Protocol Data Unit |
| RD PDU | Redirect Protocol Data Unit |

## 5.3 Protocol Data Unit Fields

| | |
|---|---|
| NPID | Network Layer Protocol Identifier |
| LI | Length Indicator |
| V/P | Version/Protocol Identifier Extension |
| TP | Type |
| CS | Checksum |
| NETL | Network entity Title Length |
| NET | Network entity Title |
| DAL | Destination Address Length |
| DA | Destination Address |
| SAL | Source Address Length |
| SA | Source Address |
| BSNPAL | SN Address Length of better route to destination |
| BSNPA | SN Address of better route to destination |

HT Holding timer

5.4 Parameters

CT Configuration Timer
RT Redirect Timer

5.5 Miscellaneous

ES End System
IS Intermediate System
SN Subnetwork
SNACP Subnetwork Access Protocol
SNICP Subnetwork Independent Convergence Protocol

## 6 Overview of the Protocol

### 6.1 Information Provided by the Protocol

This Protocol provides two types of information to Network entities which support its operation:

a) Configuration Information, and

b) Route Redirection Information

Configuration Information permits End Systems to discover the existence and reachability of Intermediate Systems and permits Intermediate Systems to discover the existence and reachability of End Systems. This information allows ESs and ISs attached to the same subnetwork to dynamically discover each other's existence and availability, thus eliminating the need for manual intervention at ESs and ISs to establish the identity of Network entities that can be used to route NPDUs.

Configuration Information also permits End Systems to obtain information about each other in the absence of an available Intermediate System.

Note:
The term "configuration information" is not intended in the broad sense of configuration as used in the context of OSI system management. Rather, only the functions specifically defined herein are intended.

Route Redirection Information allows Intermediate Systems to inform End Systems of (potentially) better paths to use when forwarding NPDUs to a particular destination. A better path could either be another IS on the same subnetwork as the ES, or the destination ES itself, if it on the same subnetwork as the source ES. Allowing the ISs to inform the ESs of routes minimizes the complexity of routing decisions in End Systems and improves performance because the ESs may

make use of the better IS or local subnetwork access for subsequent transmissions.

## 6.2 Subsets of the Protocol

A Network Entity may choose to support either the Configuration Information, the Route Redirection Information, neither, or both. If the Configuration Information is supported, it is not required that it be employed over all subnetworks to which the Network entity is attached.

## 6.3 Addressing

The Source Address and Destination Address parameters referred to in this International Standard are OSI Network Service Access Point Addresses. The syntax and semantics of an OSI Network Service Access Point Address are described in a separate document, ISO 8348/DAD2, Addendum to the Network Service Definition covering Network Layer Addressing.

## 6.4 Underlying Service Assumed by the Protocol

The underlying service required to support this protocol is defined by the primitives in Table 1.

| | | |
|---|---|---|
| SN_UNITDATA | .Request<br>.Indication | SN_Destination_Address,<br>SN_Source_Address,<br>SN_Quality_of_Service,<br>SN_Userdata |

Table 1: Service Primitives for Underlying Service

Note:
These service primitives are used to describe the abstract interface which exists between the protocol machine and an underlying real subnetwork or a Subnetwork Dependent Convergence Function which operates over a real subnetwork or real data link to provide the required underlying service.

### 6.4.1 Subnetwork Addresses

The source and destination addresses specify the points of attachment to a public or private subnetwork(s) involved in the transmission (known as Subnetwork Points of Attachment, or SNPAs).Subnetwork addresses are defined in the Service Definition of each individual subnetwork. This protocol is designed to take advantage of subnetworks which support broadcast, multicast, or other forms of multi-

destination addressing for n-way transmission. It is assumed that the SN_Destination_Address parameter may take on one of the following multi-destination addresses in addition to a normal single destination address:

All End System Network entities

All Intermediate System Network entities

Where a real subnetwork does not inherently support broadcast or other forms of transmission to multi-destination addresses, a convergence function may be used to provide n-way transmission to these multi-destination addresses.

When the SN_Destination_Address on the SN_UNITDATA.Request is a multi-destination address, the SN_Destination_Address parameter in the corresponding SN_UNITDATA.Indication shall be the same multi-destination address.

The syntax and semantics of subnetwork addresses, except for the properties described above, are not defined in this Protocol Standard.

### 6.4.2 Subnetwork User Data

The SN_Userdata is an ordered multiple of octets, and is transferred transparently between the specified subnetwork points of attachment.

The underlying service is required to support a service data unit size of at least that required to operate the Protocol for Providing the Connectionless Network Service (ISO 8473).

## 6.5 Service Assumed from Local Environment

A timer service must be provided to allow the protocol entity to schedule events.

There are three primitives associated with the S-TIMER service:

1. the S--TIMER Request,
2. the S--TIMER Response, and
3. the S--TIMER Cancel.

The S--TIMER Request primitive indicates to the local environment that it should initiate a timer of the specified name and subscript and maintain it for the duration specified by the time parameter.

The S--TIMER Response primitive is initiated by the local environment to indicate that the delay requested by the corresponding S-TIMER Request primitive has elapsed.

The S--TIMER Cancel primitive is an indication to the local environ-

ment that the specified timer(s) should be canceled.If the subscript parameter is not specified, then all timers with the specified name are canceled; otherwise, the timer of the given name and subscript is cancelled. If no timers correspond to the parameters specified, the local environment takes no action.

The parameters of the S--TIMER service primitives are specified in Table 2.

| | | |
|---|---|---|
| S--TIMER | .Request | S-Time,<br>S-Name,<br>S-Subscript |
| | .Response | S-Name,<br>S-Subscript |

Table 2: Timer Primitives

The time parameter indicates the time duration of the specified timer. An identifiying label is associated with a timer by means of the name parameter.The subscript parameter specifies a value to distinguish timers with the same name. The name and subscript taken together constitute a unique reference to the timer.

Timers used in association with a specific protocol funtion are defined under that protocol function.

Note:
This International Standard does not define specific values for the timers.Any derivations described in this Standard are not mandatory. Timer values should be chosen so that the requested Quality of Service can be provided, given the known characteristics of the underlying service.

## 6.6 Subnetwork Types

In order to evaluate the applicability of this protocol in particular configurations of End Systems, Intermediate Systems and subnetworks, three generic types of subnetwork are identified. These are:

1. the point-to-point subnetwork,

2. the broadcast subnetwork, and

3. the general topology subnetwork

These subnetwork types are discussed in the following clauses.

### 6.6.1 Point-to-Point Subnetworks

A point-to-point subnetwork supports exactly two systems. The two systems may be either two End Systems, or an End System and a single Intermediate System. A single point-to-point data link connecting two Network Entities is an example of a point-to-point subnetwork.

Configuration Information on a point-to-point Subnetwork.On a point-to-point subnetwork the Configuration Information of this protocol informs the communicating Network entities of the following:

1. Whether the topology consists only of two End Systems, or

2. One of the two systems is a Intermediate System.

Note:
On a point-to-point subnetwork, if both systems are Intermediate Systems, then this protocol is inapplicable to the situation, since a IS-to-IS protocol should be employed instead. However, there is no reason why the configuration information could not be employed in a IS-to-IS environment to ascertain the topology and initiate operation of a IS-to-IS protocol.

The Intermediate System is informed of the NSAP address(es) supported by the Network entity in the End System. This permits reachability information and routing metrics concerning these NSAPs to be disseminated to other Intermediate Systems for the purpose of calculating routes to/from this End System.

Route Redirection Information on a point-to-point Subnetwork. Route Redirection Information is not employed on point-to-point subnetworks because there are never any alternate routes.

### 6.6.2 Broadcast Subnetworks

A Broadcast subnetwork supports an arbitrary number of End Systems and Intermediate Systems, and additionally is capable of transmitting a single SNPDU to all or a subset of these systems in response to a single SN_UNITDATA.Request.An example of a broadcast subnetwork is a LAN (local area network) conforming to DIS8802/2, type 1 operation.

Configuration Information on a broadcast Subnetwork.On a broadcast subnetwork the Configuration Information of this protocol is employed to inform the communicating Network entities of the following:

1. End Systems are informed of the reachability, Network entity Title, and SNPA address(es) of each active Intermediate System on the subnetwork.

2. Intermediate Systems are informed of the NSAP addresses supported by each End System and the Subnetwork address of the ES. Once the Intermediate System obtains this information, reachability information and routing metrics concerning these NSAPs may be disseminated to other ISs for the purpose of calculating routes to/from each ES on the subnetwork.

3. In the absence of an available Intermediate System, End Systems may query over a broadcast subnetwork to discover whether a particular NSAP is reachable on the subnetwork, and if so, what SNPA address to use to reach that NSAP.

Route Redirection Information on broadcast Subnetworks.Route Redirection Information may be employed on broadcast subnetworks to permit Intermediate Systems to inform End Systems of superior routes to a destination NSAP. The superior route might be another IS on the same subnetwork as the ES, or it might be the destination ES itself, if it is directly reachable on the same subnetwork as the source ES.

### 6.6.3 General Topology Subnetworks

A general topology subnetwork supports an arbitrary number of End Systems and Intermediate Systems, but does not support a convenient multidestination connectionless transmission facility as does a broadcast subnetwork.An example of a general topology subnetwork is a subnetwork employing X.25 or ISO 8208.

Note:
The crucial distinguishing characteristic between the broadcast subnetwork and the general topology subnetwork is the "cost" of an n-way transmission to a potentially large subset of the systems on the subnetwork. On a general topology subnetwork, the cost is assumed to be close to the cost of sending an individual PDU to each SNPA on the subnetwork. Conversely, on a broadcast subnetwork the cost is assumed to be close to the cost of sending a single PDU to one SNPA on the subnetwork. Intermediate situations between these extremes are of course possible. In such cases it would be possible to treat the subnetwork as either in the broadcast or general topology categories.

Configuration Information on a general topology Subnetwork. On a general topology subnetwork the Configuration Information is generally not employed because this protocol can be very costly in the utilization (and charging for) subnetwork resources.

Route Redirection Information on a general topology Subnetwork. Route Redirection Information may be employed on general topology subnetworks to permit Intermediate Systems to inform End Systems of superior routes to a destination NSAP. The superior route might be another IS on the same subnetwork as the ES, or it might be the destination ES itself, if it is directly reachable on the same subnet-

work as the source ES.

# SECTION TWO. SPECIFICATION OF THE PROTOCOL

## 7 Protocol Functions

This section describes the functions performed as part of the Protocol. Not all of the functions must be performed by every implementation. Clause 7.12 specifies which functions may be omitted and the correct behavior where requested functions are not implemented.

### 7.1 Protocol Timers

Many of the protocol functions are timer based. This means that they are executed upon expiration of a timer rather than upon receipt of a PDU or invocation of a service primitive. The two major types of timers employed by the protocol are the Configuration Timer (CT) and the Holding Timer (HT).

#### 7.1.1 Configuration Timer

The Configuration Timer is a local timer (i.e. maintained independently by each system) which performs the Report Configuration function (see section 7.2). The timer determines how often a system reports its availability to the other systems on the same subnetwork. The shorter the Configuration Timer, the more quickly other systems on the subnetwork will become aware when the reporting system becomes available or unavailable. The increased responsiveness must be traded off against increased use of resources in the subnetwork and in the recipient systems.

#### 7.1.2 Holding Timer

The Holding Timer applies to both Configuration Information and Route Redirection Information. The value of the Holding Timer is set by the source of the information and transmitted in the appropriate PDU. The recipient of the information is expected to retain the information no longer than the Holding Timer. Old Configuration or Route Redirection information must be discarded after the Holding Timer expires to ensure the correct operation of the protocol.

Further discussion of the rationale for these timers and guidelines for their use may be found in annex 10.

### 7.2 Report Configuration Function

The Report Configuration Function is used by End Systems and Intermediate Systems to inform each other of their reachability and current subnetwork address. This function is invoked every time the local Configuration Timer (CT) expires in an ES or IS. It is also invoked upon receipt of a Query Configuration PDU from another End System.

### 7.2.1 Report Configuration by End Systems

An End System constructs and transmits one ESH PDU (ESH stands for "End System Hello") for each NSAP it serves, and issues one SN_UNITDATA.- Request with the ESH PDU as the SNSDU on each subnetwork to which it is attached.

Note:
The necessity to transmit a separate ESH PDU for each NSAP served by the Network entity arises from the lack of a formalized relationship between Network Entity Titles and NSAP addresses. If this relationship could be constrained to require that all NSAP addresses be assigned as leaf subdomains of a domain represented by the local Network entity's Network entity Title, then a single ESH PDU could be transmitted containing the ESs Network entity Title.The Network entity Title would then imply which NSAPs might be present at that End system.

The Holding Timer (HT) field is set to approximately twice the ESs Configuration Timer (CT) parameter. This variable is set to a value large enough so that even if every other ESH PDU is discarded (due to lack of resources), or otherwise lost in the subnetwork, the configuration information will still be maintained. The value must be set small enough so that Intermediate Systems can respond in a timely fashion to End Systems becoming available or unavailable.

The SN_Destination_Address parameter is set to the group address that indicates "All Intermediate System Network Entities". This ensures that a single transmission on a broadcast subnetwork will reach all of the active Intermediate Systems.

Note:
The actual value of the SN_Destination_Address used to mean "All Intermediate System Network Entities" is subnetwork dependent and will most likely vary from subnetwork to subnetwork. It would of course be desirable that on widely-used subnetwork types (such as those based on DIS 8802) that this value and the value of the "All End System Network Entities" group address, be standardized.

### 7.2.2 Report Configuration by Intermediate Systems

An Intermediate System constructs a single ISH PDU (ISH stands for "Intermediate System Hello") containing the ISs Network Entity Title and issues one SN_UNITDATA.Request with the ISH PDU as the SNSDU on each subnetwork to which it is attached.

The Holding Timer (HT) field is set to approximately twice the Intermediate System's Configuration Timer (CT) parameter. This variable is set to a value large enough so that even if every other ISH PDU is discarded (due to lack of resources), or otherwise lost in the subnetwork, the configuration information will still be maintained.The value must be set small enough so that End Systems will quickly cease

to use ISs that have failed, thus preventing "black holes" in the Network.

The SN_Destination_Address parameter is set to the group address that indicates "All End System Network Entities".This ensures that a single transmission on a broadcast subnetwork will reach all of the active End Systems.

## 7.3 Record Configuration Function

The Record Configuration function receives ESH or ISH PDUs, extracts the configuration information, and adds or replaces the corresponding configuration information in the local Network entity's Routing Information base. If insufficient space is available to store new configuration information, the PDU is discarded. No Error Report is generated.

Note:

The protocol is described such that End Systems receive and record only ISH PDUs and Intermediate Systems receive and process only ESH PDUs. If an ES so desires however, it may decide to process ESH PDUs as well (on a broadcast network this is easily done by enabling the appropriate group address). There is potentially some performance improvement to be gained by doing this, at the expense of extra memory, and possibly extra processing cycles in the End System.The ES, by recording other ESs' Configuration information, may be able to route NPDUs directly to ESs on the local subnetwork without first being redirected by a Intermediate System.

Similarly, Intermediate Systems may choose to receive the ISH PDUs of other ISs, allowing this protocol to be used as the initialization and topology maintenance portion of a full IS-to-IS routing protocol. Both of these possibilities are for further study.

## 7.4 Flush Old Configuration Function

The Flush Old Configuration Function is executed to remove Configuration entries in the routing information base whose Holding Timer has expired. When the Holding Time for an ES or IS expires, this function removes the corresponding entry from the routing information base of the local Network Entity.

## 7.5 Query Configuration Function

The Query Configuration Function is performed under the following circumstances:

1. The End System is attached to a broadcast subnetwork,

2. There is no Intermediate System currently reachable on the subnetwork (i.e. no ISH PDUs have been received since the last

information was flushed by the Flush Old Configuration Function),

3. The Network Layer's Route PDU Function needs to obtain the SNPA address to which to forward a PDU destined for a certain NSAP, and

4. The SNPA address cannot be obtained either by a local transformation or a local table lookup.

Note:
Despite appearances, this is actually a quite common case, since it is likely that there will be numerous isolated Local Area Networks without Intermediate Systems to rely upon for obtaining routing information (e.g.via the Request Redirect Function of this protocol). Further, if the Intermediate System(s) are temporarily unavailable, without this capability communication on the local subnetwork would suffer unless manually-entered tables were present in each End System or all NSAPs of the subnetwork had the subnetwork SNPA address embedded in them.

The End System, when needing to route an NPDU to a destination NSAP whose SNPA is unknown issues an SN_UNITDATA.Request with the NPDU as the SN_Userdata.The SN_Destination_Address parameter is set to the group address that indicates "All End System Network Entities".

Subsequently an ESH PDU may be received containing the NSAP address along with the corresponding SNPA address (see clause 7.6). In such a case the End System executes the Record Configuration function for the NSAP, and therefore will be able to route subsequent PDUs to that destination using the specified SNPA. If no ESH PDU is received, the End System may declare the destination NSAP is not reachable. The length of time to wait for a response before indicating a failure or the possibility of repeating the process some number of times before returning a failure are local matters and are not specified in this standard.

## 7.6 Configuration Response Function

The Configuration Response function is performed when an End System attached to a broadcast subnetwork receives an NPDU addressed to one of its NSAPs, with the SN_Destination_Address from the SN_UNITDATA.Indication set to the group address "All End System Netowrk Entities". This occurs as a result of another ES having performed the Query Configuration function described in clause 7.5.

The End System constructs an ESH PDU identical in content to the ESH PDU constructed by the Report Configuration function (see clause 7.2.1) for the NSAP to which the received NPDU was addressed.It then transmits the ESH PDU to the source of the original NPDU by issuing an SN_UNITDATA.Request with the SN_Destination_Address set to the value of the SN_Source_Address received in the SN_UNITDATA.Indication with the original NPDU.

## 7.7 Request Redirect Function

The Request Redirect Function is present only in Intermediate Systems and is closely coupled with the Routing and Relaying Functions of Intermediate Systems. The Request Redirect Function is coupled with the "Route PDU Function" described in clause 6.5 of ISO 8473. The Request Redirect Function is performed after the Route PDU function has calculated the next hop of the Data PDU's path.

When an NPDU is to be forwarded by a Intermediate System, the Request Redirect Function first examines the SN_Source_Address associated with the SN_UNITDATA.Indication which received the SNSDU (containing this NPDU). If the SN_Source_Address is not from an End System on the local subnetwork (determined by examining the Configuration information obtained through the Record Configuration Function), then this function does no further processing of the NPDU.

If the NPDU was received directly from an ES the output of the ISs Routing and Relaying function for this NPDU is examined. This output will contain, among other things, the following pieces of information:

1. a local identifier for the subnetwork over which to forward the NPDU, plus either

2. the Network entity title and subnetwork address of the IS to which to forward the NPDU, or

3. the subnetwork address of the destination End System.

The Request Redirect function must now determine if the source ES could have sent the NPDU directly to the Network entity the Intermediate System is about to forward the PDU to. If any of the following conditions hold, the source ESshould be informed of the "better" path (by sending an RD PDU to the originating ES):

1. The next hop is to the destination system, and the destination is directly reachable (at subnetwork address BSNPA) on the source ESs subnetwork, or

2. The next hop is to a Intermediate System which is connected to the same subnetwork as the ES.

If the better path exists, the IS first completes normal processing of the received NPDU and forwards it.It then constructs a Redirect PDU (RD PDU) containing the Destination Address of the original NPDU, the subnetwork address of the better next hop (BSNPA), the Network Entity Title of the IS to which the ES is being redirected (unless the redirect is to the destination ES), a Holding Time (HT), QoS Maintenance, Priority, and Security options that were present in the Data NPDU (these are simply copied from the Data PDU). The HT is set

to the value of the local Redirect Timer (RT). See Annex A for a discussion of how to choose the value of RT. If there are insufficient resources to both forward the original NPDU and to generate and send an RD PDU, the original NPDU must be given preference. The Intermediate System (assuming it has sufficient resources) then sends the RD PDU to the source End System using the SN_Source_Address of the received NPDU as the SN_Destination_Address for the SN_UNITDATA.-Reqeust.

7.8 Record Redirect Function

The Record Redirect Function is present only in End Systems. This function is invoked whenever an RD PDU is received. It extracts the redirect information and adds or replaces the corresponding redirection information in the local Network entity's Routing Information base. The essential information is the redirection mapping from a Destination Address to a subnetwork address, along with the Priority, Security, and QoS Maintenance options and the Holding Time for which this mapping is to be considered valid. If the Redirect was to another Intermediate System, the Network Entity Title of the IS is recorded as well.

Note:
If insufficient memory is available to store new redirection information, the RD PDU may be safely discarded since the original Intermediate System will continue to forward PDUs on behalf of this Network entity anyway.

7.9 Refresh Redirect Function

The Refresh Redirect Function is present only in End Systems. This function is invoked whenever an NPDU is received by a destination ES. It is closely coupled with the function that processes received NPDUs at a destination Network Entity.This is the "PDU Decomposition" function in ISO 8473. The purpose of this function is to increase the longevity of a redirection without allowing an incorrect route to persist indefinitely. The Source Address (SA), Priority, Security, and QoS options are extracted and compared to any Destination Address and QoS parameters being maintained in the Routing Information base (such information would have been stored by the Record Redirect Function). If a corresponding entry is found, the previous hop of the PDU is obtained from the SN_Source_Address parameter of the SN_Unitdata.Indication primitive by which it was received. If this address matches the next hop address stored with the redirection information, the remaining holding time for the redirection is reset to the original holding timer that was obtained from the RD PDU.

Note:
The purpose of this function is to avoid timing out redirection entries when the Network entity is receiving return traffic from the destination via the same path over which it is currently sending traffic.This is

particularly useful when the destination system is on the same subnetwo as the source, since after one redirect no IS need be involved in the ES-to-ES traffic.

This function must operate in a very conservative fashion however, to prevent the formation of black holes. The remaining holding time should be refreshed only under the exact conditions specified above. For a discussion of the issues surrounding the refresh of redirection information, see Annex 10.

## 7.10 Flush Old Redirect Function

The Flush Old Redirect Function is executed to remove Configuration entries in the routing information base whose Holding Timer has expired. When the Holding Time for an ES or IS expires, this function removes the corresponding entry from the routing information base of the local Network Entity.

## 7.11 PDU Header Error Detection

The PDU Header Error Detection function protects against failure of Intermediate or End System Network entities due to the processing of erroneous information in the PDU header.The function is realized by a checksum computed on the entire PDU header. The checksum is verified at each point at which the PDU is processed. If the checksum calculation fails, the PDU must be discarded.

The use of the Header Error Detection function is optional and is selected by the originating Network Entity. If the function is not used, the checksum field of the PDU header is set to zero.

If the function is selected by the originating Network Entity, the value of the checksum field causes the following formulf to be satisfied:

(The Sum from i=1 to L of a(i)) (mod 255) = 0

(The Sum from i=1 to L of (L - i + 1) * a(i)) (mod 255) = 0

where L = the number of octets in the PDU header, and a(i) = the value of the octet at position i. The first octet in the PDU header is considered to occupy position i = 0.

When the function is in use, neither octet of the checksum field may be set to zero.

### 7.12 Classification of Functions

Implementations do not have to support all of the functions described in clause 7. Functions are divided into four categories:

Type A: These functions must be supported in all cases.

Type B: These functions must be supported by Systems which implement the Configuration Information.

Type C: These functions must be supported by Systems which implement the Redirect Information.

Type D: These functions are optional.

If a PDU is received which invokes an optional function that is not implemented, that PDU is discarded.

Table 3 shows how the functions are divided into these four categories, and to which type of system (ES, IS, or both) they apply.

| Function | Category | System Type |
|---|---|---|
| Report Configuration | B | ES,IS |
| Record Configuration | B | ES,IS |
| Configuration Response | A | ES |
| Flush Old Configuration | B | ES,IS |
| Request Redirect | C | IS |
| Query Configuration | B | ES |
| Record Redirect | C | ES |
| Refresh Redirect | D | ES |
| Flush Old Redirect | C | ES |
| PDU Header Error Detection | A | ES,IS |

Table 3: Categories of Protocol Functions

## 8 Structure and Encoding of PDUs

Note:
The encoding of the PDUs for this protocol is compatible with that used in ISO 8473.

Temporary Note:
The method employed for describing the encoding of PDUs is provisional. Member bodies are requested to comment on whether another method (such as ASN.1 with an appropriate concrete syntax) would be preferable.

## 8.1 Structure

All Protocol Data Units shall contain an integral number of octets.The octets in a PDU are numbered starting from one (1) and increasing in the order in which they are put into an SNSDU. The bits in an octet are numbered from one (1) to eight (8), where bit one (1) is the low-order bit. When consecutive octets are used to represent a binary number, the lower octet number has the most significant value.

Any subnetwork supporting this protocol is required to state in its specification the way octets are transferred, using the terms "most significant bit" and "least significant bit". The PDUs of this protocol are defined using the terms "most significant bit" and "least significant bit".

Note:

When the encoding of a PDU is represented using a diagram in this section, the following representation is used:

a) octets are shown with the lowest numbered octet to the left, higher number octets being further to the right;
b) within an octet, bits are shown with bit eight (8) to the left and bit one (1) to the right.

PDUs shall contain, in the following order:

1. the fixed part;
2. the Network address part;
3. the Subnetwork address part, if present; and
4. the Options part, if present.

## 8.2 Fixed Part

### 8.2.1 General

The fixed part contains frequently occurring parameters including the type code (ESH, ISH, or RD) of the protocol data unit.The length and the structure of the fixed part are defined by the PDU code.

The fixed part has the following format:

| | | | | Octet |
|---|---|---|---|---|
| Network Layer Protocol Identifier | | | | 1 |
| Length Indicator | | | | 2 |
| Version/Protocol Id Extension | | | | 3 |
| reserved (must be zero) | | | | 4 |
| 0 | 0 | 0 | Type | 5 |
| Holding Time | | | | 6,7 |
| Checksum | | | | 8,9 |

Figure 1: PDU Header -- Fixed Part

8.2.2 Network Layer Protocol Identifier

The value of this field shall be 1000 0010.

Temporary Note:
The value 1000 0010 is provisional, pending resolution of the NLPID issue in SC6.

This field identifies this Network Layer Protocol as ISO SC6/N4053, End System to Intermediate System Routing Exchange Protocol for use in conjunction with ISO 8473.

8.2.3 Length Indicator

The length is indicated by a binary number, with a maximum value of 254 (1111 1110).The length indicated is the length of the entire PDU (which consists entirely of header, since this protocol does not carry user data) in octets, as described in clause 8.1. The value 255 (1111 1111) is reserved for possible future extensions.

8.2.4 Version/Protocol Identifier Extension

The value of this field is binary 0000 0001. This identifies a standard version of ISO xxxx, End System to Intermediate System Routing Exchange Protocol for use in conjunction with ISO 8473.

8.2.5 Type Code

The Type code field identifies the type of the protocol data unit. Allowed values are given in table 4.

| | Bits | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|
| ESH PDU | | 0 | 0 | 0 | 1 | 0 |
| ISH PDU | | 0 | 0 | 1 | 0 | 0 |
| RD PDU | | 0 | 0 | 1 | 1 | 0 |

Table 4: Valid PDU Types

All other PDU type values are reserved.

8.2.6 Holding Time

The Holding Time field specifies for how long the receiving Network entity should retain the configuration/routing information contained in this PDU. The receiving Network entity should discard any information obtained from this PDU from its internal state when the holding time expires. The Holding time field is encoded as an integral number of micro-fortnights.

8.2.7 PDU Checksum

The checksum is computed on the entire PDU header. A checksum value of zero is reserved to indicate that the checksum is to be ignored. The operation of the PDU Header Error Detection function (Clause 7.11) ensures that the value zero does not represent a valid checksum. A non-zero value indicates that the checksum must be processed. If the checksum calculation fails, the PDU must be discarded.

8.3 Network Address Part

8.3.1 General

Address parameters are distinguished by their location. The different PDU types carry different address parameters however.The ESH PDU carries a Source NSAP address (SA); the ISH PDU carries a Intermediate System Network entity Title (NET); and the RD PDU carries a Destination NSAP address (DA), and possibly a Network Entity Title (NET).

8.3.2 NPAI (Network Protocol Address Information) Encoding

The Destination and Source Addresses are Network Service Access Point

addresses as defined in ISO 8348/AD2, Addendum to the Network Service Definition Covering Network Layer addressing.The Network Entity Title address parameter is defined in clause 4.5. The Destination Address, Source Address, and Network Entity Title are encoded as NPAI using the binary syntax defined in clause 8.3.1 of ISO 8348/AD2.

The address information is of variable length. Each address parameter is encoded as follows:

| | |
|---|---|
| Octet n | Address parameter Length Indicator (e.g., 'm') |
| Octets n + 1 thru n + m | Address Parameter Value |

Figure 2: Address Parameters

8.3.3 Source Address Parameter for ESH PDU

The Source Address is the NSAP address of an NSAP served by the Network entity sending the ESH PDU. It is encoded in the ESH PDU as follows:

| | Octet |
|---|---|
| Source Address Length Indicator (SAL) | 10 |
| : Source Address (SA) : | 11 |
| | m - 1 |

Figure 3: ESH PDU - Network Address Part

8.3.4 Network Entity Title Parameter for ISH PDU

The Network entity Title parameter is the Network Entity Title of the Intermediate System sending the ISH PDU. It is encoded in the ISH PDU as follows:

| | Octet |
|---|---|
| Network Entity Title Length Indicator (NETL) | 10 |
| Network Entity Title (NET) | 11 |
| | m - 1 |

Figure 4: ISH PDU - Network Address Part

### 8.3.5 Destination Address Parameter for RD PDU

The Destination Address is the NSAP address of a destination associated with some NPDU being forwarded by the Intermediate System sending the RD PDU. It is encoded in the RD PDU as follows:

| | Octet |
|---|---|
| Destination Address Length Indicator (DAL) | 10 |
| Destination Address (DA) | 11 |
| | m - 1 |

Figure 5: RD PDU - Network Address Part

## 8.4 Subnetwork Address Part

The Subnetwork Address Part is present only in RD PDUs.It is used to indicate the subnetwork address of another Network entity on the same subnetwork as the End System (and Intermediate System) which may be a better path to the destination specified in the Network Address Part. The Subnetwork Address parameter is encoded in the same manner as the Network Address parameters.

### 8.4.1 Subnetwork Address Parameter for RD PDU

The Subnetwork Address Parameter is encoded in the RD PDU as follows:

| | Octet |
|---|---|
| Subnetwork Address Length Indicator (BSNPAL) | m |
| Subnetwork Address (BSNPA) | m + 1<br>n - 1 |

Figure 6: ESH PDU - Address Part

## 8.5 Options Part

### 8.5.1 General

The options part is used to convey optional parameters. The options part
of the PDU header is illustrated below:

| | Octet |
|---|---|
| Options | p<br>q |

Figure 7: All PDUs - Options Part

If the options part is present, it may contain one or more parameters. The number of parameters that may be contained in the options part is constrained by the length of the options part, which is determined by the following formula:

PDU Header Length - (length of fixed part + length of address part + length of segmentation part),

and by the length of the individual optional parameters.

Parameters defined in the options part may appear in any order. Duplication of options is not permitted.Receipt of a PDU with an option duplicated must be treated as a protocol error.

The encoding of parameters contained within the options part of the PDU header is illustrated below in figure 8.

Octets

| | |
|---|---|
| n | Parameter Code |
| n + 1 | Parameter Length |
| n + 2 to n + m + 1 | Parameter Value |

Figure 8: Encoding of Option Parameters

The parameter code field is coded in binary and, without extensions, provides a maximum of 255 different parameters. No parameter codes use bits 8 and 7 with the value 00, so the actual maximum number of parameters is lower. A parameter code of 255 (binary 1111 1111) is reserved for possible future extensions.

The parameter length field indicates the length, in octets, of the parameter value field.The length is indicated by a positive binary number, m, with a theoretical maximum value of 254. the practical maximum value of m is lower. For example, in the case of a single parameter contained within the options part, two octets are required for the parameter code and the parameter length indicators. Thus, the value of m is limited to:

m = 252-(length of fixed part +length of address part +length of segmentation part )

For each succeeding parameter the maximum value of m decreases. The parameter value field contains the value of the parameter identified in the parameter code field.

The following parameters are permitted in the options part.

### 8.5.2 Security

The Security parameter conveys information about the security requested in the Data PDU that caused the containing RD PDU to be generated. This parameter has the same encoding and semantics as the Security parameter in ISO 8473.

Parameter Code: 1100 0101

Parameter Length: variable

Parameter Value: See Section 7.5.3 of ISO 8473

### 8.5.3 Quality of Service Maintenance

The Quality of Service parameter conveys information about the quality of service requested in the Data PDU that caused the containing RD PDU to be generated.

This parameter has the same encoding and semantics as the QoS Maintenance parameter in ISO 8473.

Parameter Code: 1100 0011

Parameter Length: variable

Parameter Value: See Section 7.5.6 of ISO 8473

### 8.5.4 Priority

The Priority parameter conveys information about the priority requested in the Data PDU that caused the containing RD PDU to be generated.

This parameter has the same encoding and semantics as the Priority parameter in ISO 8473.

Parameter Code: 1100 1101

Parameter Length: one octet

Parameter Value: See Section 7.5.7 of ISO 8473

## 8.6 End System Hello (ESH) PDU

### 8.6.1 Structure

The ESH PDU has the following format:

| | Octet |
|---|---|
| Network Layer Protocol Identifier | 1 |
| Length Indicator | 2 |
| Version/Protocol Id Extension | 3 |
| reserved (must be zero) | 4 |
| 0 0 0 Type | 5 |
| Holding Time | 6,7 |
| Checksum | 8,9 |
| Source Address Length Indicator (SAL) | 10 |
| Source Address (SA) | 11 |
| | m - 1 |
| Options | m |
| | p - 1 |

Figure 9: ESH PDU Format

## 8.7 Intermediate System Hello (ISH) PDU

### 8.7.1 Structure

The ISH PDU has the following format:

| | | | | | Octet |
|---|---|---|---|---|---|
| Network Layer Protocol Identifier | | | | | 1 |
| Length Indicator | | | | | 2 |
| Version/Protocol Id Extension | | | | | 3 |
| reserved (must be zero) | | | | | 4 |
| 0 | 0 | 0 | Type | | 5 |
| Holding Time | | | | | 6,7 |
| Checksum | | | | | 8,9 |
| Network Entity Title Length Indicator (NETL) | | | | | 10 |
| : Network Entity Title (NET) : | | | | | 11<br>m - 1 |
| : Options : | | | | | m<br>p - 1 |

Figure 10: ISH PDU Format

## 8.8 Redirect (RD) PDU

### 8.8.1 Structure

The RD PDU has the following format:

| | | | | Octet |
|---|---|---|---|---|
| Network Layer Protocol Identifier | | | | 1 |
| Length Indicator | | | | 2 |
| Version/Protocol Id Extension | | | | 3 |
| reserved (must be zero) | | | | 4 |
| 0 | 0 | 0 | Type | 5 |
| Holding Time | | | | 6,7 |
| Checksum | | | | 8,9 |
| Destination Address Length Indicator (DAL) | | | | 10 |
| : Destination Address (DA) : | | | | 11<br>m - 1 |
| Subnetwork Address Length Indicator (BSNPAL) | | | | m |
| : Subnetwork Address (DBSNPA) : | | | | m + 1<br>n - 1 |
| Network Entity Title Length Indicator (NETL) | | | | n |
| : Network Entity Title (NET) : | | | | n + 1<br>p - 1 |
| : Options : | | | | p<br>q - 1 |

Figure 11: RD PDU Format when Redirect is to an IS

| | Octet |
|---|---|
| Network Layer Protocol Identifier | 1 |
| Length Indicator | 2 |
| Version/Protocol Id Extension | 3 |
| reserved (must be zero) | 4 |
| 0 0 0 Type | 5 |
| Holding Time | 6,7 |
| Checksum | 8,9 |
| Destination Address Length Indicator (DAL) | 10 |
| : Destination Address (DA) : | 11<br>m - 1 |
| Subnetwork Address Length Indicator (BSNPAL) | m |
| : Subnetwork Address (DBSNPA) : | m + 1<br>n - 1 |
| NETL = 0 | n |
| : Options : | n + 1<br>p - 1 |
| Quality of Service | n + 1 |

Figure 12: RD PDU Format when Redirect is to an ES

## 9 Formal Description

{Maybe next pass...}

## 10 Conformance

See Clause 6.2.

# ANNEX A. SUPPORTING TECHNICAL MATERIAL

## A.1 Use of Timers

This protocol makes extensive use of timers to ensure the timeliness and accuracy of information disseminated using the Configuration and Route Redirection functions.This section discusses the rationale for using these timers and provides some background for how they operate.

Systems using this protocol learn about other systems exclusively by receiving PDUs sent by those systems. In a connectionless environment, a system must periodically receive updated information to ensure that the information it previously received is still correct. For example, if a system on a subnetwork becomes unavailable (either it has ceased operating, or its SNPA becomes inoperative) the only way another system can detect this fact is by the absence of transmissions from that system. If information were retained in the absence of new PDUs being received, configuration and/or routing information would inevitably become incorrect. The Holding Timers specified by this protocol guarantee that old information will not be retained indefinitely.

A useful way of thinking of the configuration and route redirection information is as a cache maintained by each system. The cache is periodically flushed to ensure that only up-to-date information is stored.Unlike most caches, however, the time to retain information is not a purely local matter. Rather, information is held for a period of time specified by the source of the information. Some examples will help clarify this operation.

### A.1.1 Example of Holding Time for Route Redirection

Route Redirection Information is obtained by an End System through the Request Redirect function (see clause 7.7).It is quite possible that a Intermediate System might redirect an End System to another IS which has recently become unavailable (this might happen if the IS-to-IS routing algorithm is still converging following a configuration change). If the Holding Timer were not present, or was set very long by the sending IS, an End System would have been redirected into a Black Hole from which none of its Data PDUs would ever emerge. The length of the Holding Timer on Redirects specifies, in essence, the length of time black holes are permitted to exist.

On the other hand, setting the Holding Timer on Route Redirects very short to minimize the effect of black holes has other undesirable consequences.First, for each PDU that causes a redirect, an additional PDU beside the original Data PDU must be composed and transmitted; this increases overhead. Second, each time a "working" redirect's Holding Timer expires, the redirected End System will revert to a poorer route for at least one PDU.

### A.1.2 Example of Holding Timer for Configuration Information

A similar type of problem can occur with respect to Configuration information. If the Holding Time of a ISH PDU (see clause 7.2.2) is set very long, and the only Intermediate System (which has been sending this Configuration Information) on the subnetwork becomes unavailable, a subnetwork-wide black hole can form. During this time, End Systems on the subnetwork may not be able to communicate with each other because they presume that a Intermediate System is operating which will forward their Data PDUs to destination ESs on the local subnetwork and return RD PDUs.Once the Holding Time expires, the ESs will realize that no IS is available and will take their only recourse, which is to send their traffic directly on the local subnetwork.

Given the types of problems that can occur, it is important that responsibility for incorrect information can be unambiguously assigned to the source of the information. For this reason all Holding Timers are calculated by the source of the Configuration or Route Redirection information and communicated explicitly to each recipient in the appropriate PDU.

## A.2 Refresh and timeout of Redirection information

The protocol allows End Systems to refresh redirection information without first allowing the holding time to expire and being redirected by a Intermediate System for a second (or subsequent) time. Such schemes are prevalent in connectionless subnetworks and are often called "reverse path information", "previous hop cache" or something similar.

Refreshing the redirection information has obvious performance benefits, but can be dangerous if not handled in a very conservative fashion. In order for a redirection to be safely refreshed, all of the following conditions must hold:

1. The source address of the received PDU must be exactly the same as the destination address specified in a prior RD PDU (this defines a "match" on the redirection information). Making assumptions about the equivalence of abbreviated addresses, group addresses, or similar "special" addresses is dangerous since routing for these addresses cannot be assumed to be the same.

2. The Quality of Service parameters of the received PDU must be exactly the same as the QoS parameters specified in the matching (by destination address) redirection entry.Again, there is no guarantee that PDUs with different QoS parameters will be routed the same way. It is quite possible that the redirected path is even a black hole for certain values of the QoS parameters (the security field is a good example).

3. The "previous hop" of the received Data PDU must match the "next hop" stored in the redirection information. Specifically, the SN_Source_Address of the SN_UNITDATA.Indication which received the PDU must match exactly the SN_Destination_Address specified in the redirect to be used for sending traffic via the SN_UNITDATA.Request primitive. This comparison ensures that redirects are refreshed only when the reverse traffic is being received from the same IS (or destination ES) as the forward traffic is being sent through (or to). This check make certain that redirects are not refreshed for just on the basis of traffic being received from the destination. It is quite possible that the traffic is simply indicating that the forward path in use is not working!

Note that these conditions still allow refresh in the most useful and common cases where either the destination is another ES on the same subnetwork as the source ES, or the redirection is to a IS which is passing traffic to/from the destination in both directions (i.e. the path is symmetric).

## A.3 System Initialization Considerations

This protocol is designed to make the exchange of information as free as possible from dependencies between the two types of systems. therefore, it is not possible for an End System to request all Intermediate Systems on a subnetwork to report their configuration, nor is it possible for an Intermediate System to request all End Systems on a subnetwork to report their configuration.

In certain operating environments a constraint may be imposed than an ES, upon becoming operational, must discover the existence of an IS as soon as possible.The converse relationship also holds if it is necessary for an IS to discover the existence of End Systems as soon as possible. In both cases the availability of this information is normally determined by the Configuration Timer of the system for which the knowledge is desired. there is therefore a tradeoff between the overhead associated with performing the Report and Record Configuration functions and the timely availability of the configuration information. Decreasing the Configuration Timer increases the availability at the expense of an increase in overhead.

The following solution is recommended for addressing the constraint described above. When the Record Configuration function is invoked in either an End System or an Intermediate System, the function will determine if the received configuration information was previously unknown.If this is the case, then the Report Configuration function may be invoked before the expiration of the system's Configuration Timer. The Hello PDU generated by the Report Configuration function is then sent only to the Network Entity whose configuration was previously unknown. Thus when an ES or IS first becomes operational it immediately reports its configuration. As soon as systems of the other type discover the new network entity, they will make their own

configuration known to this entity.

The additional overhead incurred by this solution is minimal. Also, since the discovery of new configurations is made timely by this approach the Configuration Timer period can be increased in order to decrease the overhead of the configuration functions, provided that other factors not discussed here are accounted for by the longer time period.One caveat is that the first Hello PDU generated by a system may be lost during transmission. To solve this problem one or more additional PDUs may be transmitted at short time intervals during this initialization period.

Note that this solution may be implemented in ISs only, in ESs only, or in both Intermediate and End Systems.This decision is purely a local matter and may be alterable through System Management.

## A.4 Optimizations for Flushing Redirects

An ES will attempt to forward NPDUs through an IS to which it has been redirected until the Holding Timer specified in the RD PDU has expired, even if that IS is no longer reachable. Under certain circumstances, it is possible to do better and recognize the existence of a black hole sooner. In particular, if the ES expects to hear ISH PDUs from the IS to which it has been redirected, and the Holding Timer for that IS expires, all knowledge of the IS may be forgotten by the ES. This includes any redirects, which may be flushed (see the Flush Old Redirect function) even though their timeouts have not expired.